

APPENDIX E

File Handling in C

1. Write a program to store records of an employee in employee file. The data must be stored using binary file.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    typedef struct employee
    {
        int emp_code;
        char name[20];
    };
    FILE *fp;
    struct employee e[2];
    int i;
    fp = fopen("employee.txt", "wb");
    if(fp==NULL)
    {
        printf("\n Error opening file");
        exit(1);
    }
    printf("\n Enter the details employees");
    for(i = 0; i < 2; i++)
    {
        printf("\n\n Enter the employee code:");
        scanf("%d", &e[i].emp_code);
        printf("\n\n Enter the name of the employee: ");
        scanf("%s", e[i].name);
        fwrite(&e[i], sizeof(e[i]), 1, fp);
    }
    fclose(fp);
    getch();
    return 0;
}
```

Output

```
Enter the details of employees
Enter the employee code: 01
Enter the name of the employee: Gargi
Enter the employee code: 02
Enter the name of the employee: Nikita
```

2. Write a program to read the records stored in 'employee.txt' file in binary mode.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
{
    typedef struct employee
    {
        int emp_code;
        char name[20];
    };
    FILE *fp;
    struct employee e;
    int i;
    clrscr();
    fp = fopen("employee.txt", "rb");
    if(fp==NULL)
    {
        printf("\n Error opening file");
        exit(1);
    }
    printf("\n THE DETAILS OF THE EMPLOYEES ARE ");
    while(1)
    {
        fread(&e, sizeof(e), 1, fp);
        if(feof(fp))
            break;
        printf("\n\n Employee Code: %d", e.emp_code);
        printf("\n\n Name: %s", e.name);
    }
    fclose(fp);
    getch();
    return 0;
}
```

Output

```
Employee Code: 01
Name: Gargi
Employee Code: 02
Name: Nikita
```